Bayesian Methods for Assessing System Reliability: Models and Computation

Todd L. Graves and Michael Hamada

Statistical Sciences

Los Alamos National Laboratory

February 17, 2005 0935

Abstract

There are many challenges with assessing the reliability of a system today. These challenges arise because a system may be aging and full system tests may be too expensive or can no longer be performed. Without full system testing, one must integrate (1) all science and engineering knowledge, models and simulations, (2) information and data at various levels of the system, e.g., subsystems and components and (3) information and data from similar systems, subsystems and components. The analyst must work with various data types and how the data are collected, account for measurement bias and uncertainty, deal with model and simulation uncertainty and incorporate expert knowledge.

Bayesian hierarchical modeling provides a rigorous way to combine information from multiple sources and different types of information. However, an obstacle to applying Bayesian methods is the need to develop new software to analyze novel statistical models. We discuss a new statistical modeling environment, YADAS, that facilitates the development of Bayesian statistical analyses. It includes classes that help analysts specify new models, as well as classes that support the creation of new analysis algorithms. We illustrate these concepts using several examples.

1 Challenges in Modern Reliability Analyses

There are many challenges with assessing the reliability of a system today. First, full system testing may be prohibitively expensive or even prohibited. For this reason and others, it is important to be able to make use of expert opinion and information in the form of physics/engineering/material science based models (deterministic and stochastic) or simulation and account for model bias and uncertainty. Results from multiscale science/engineering experiments also need to incorporated. One must be able to handle complex system reliability models, including reliability block diagrams, fault trees and networks. One must incorporate data at various levels (system, subsystem, component), and properly account for how higher level event data inform about lower level data. In this context, models for subsystems or even components can be non-trivial. The effects of aging and other covariates including those that define subpopulations, are often of interest. Efficient analysis entails combining information and data from similar systems, subsystems and components. Reliability data can come in various flavors, including binomial counts, Poisson counts, failure times, degradation data, and accelerated reliability data. Such data may be non-trivial to analyze in their own right. How such data are collected must also be considered. For example, measurement error (bias and precision) from destructive or nondestructive evaluation techniques may be too large to ignore.

These challenges go beyond that addressed in the system reliability literature (Cole (1975), Mastran (1976), Mastran and Singpurwalla (1978), Natvig and Eide (1987), Martz, Waller and Fickas (1988), Martz and Waller (1990)) which mostly consider binomial data and except for the last two references which consider combining multi-level binomial data. This literature also predates the advances made in Bayesian computation in the 1990's and resorts to various approximations. Such approximations will be hard to generalize for these challenges.

An outline of this paper is as follows. First we consider three examples which illustrate these challenges. Next we discuss a statistical modeling environment which can support the needs of modern reliability analyses. Then we return to the three examples and present results of their reliability analyses. Finally, we conclude with a discussion.

2 Three Important Examples

We discuss three examples of challenging statistical problems that arise in reliability estimation. First, even the analysis of a single component can require development of new techniques. Consider the case in which there are indications that a component's manufacturing lot impacts its reliability, and some of the test data are obtained in ways that might favor the sampling of (un)reliable items. Second, we discuss the estimation of the reliability of a system based on (1) system tests, where failures provide partial information about which components may have failed, and (2) specification tests, which measure whether components meet specifications that relate imperfectly to system success. Finally, we present an ambitious approach to integrating many sorts of component data into a system reliability analysis.

2.1 Example 1: Reliability of a Component Based on Biased Sampling

Our first example, which deals with reliability estimation for a single component, is discussed in Graves *et al* (2004). Of interest is the prevalence of a certain feature in an existing population of items. Some items have already been destructively tested and removed from the population. There is reason to believe that the probability that an item has the feature is related to the lot in which it was manufactured, but it is not obviously appropriate to assume that the feature is confined to a small number of lots. We handle this situation with a Bayesian hierarchical model, $p_i \sim \text{Beta}(a, b)$, where p_i is the probability that an item in lot i was manufactured with the feature, and where a and b are given prior distributions, so that a test on an item in one lot is informative about the prevalence of the feature in the other lots, but more informative about its own lot. A further complication is we are not willing to assume that the process by which items were selected for sampling was done so that items with and without the feature were equally likely to be sampled. (We do also have some truly random samples along with these "convenience samples.") Naive estimation is therefore in danger of systematically over- or under-estimating the prevalence. We use the *extended hypergeometric* distribution (see Graves *et al* (2004) and its references) to allow for the possibility of biased sampling; using this distribution for this purpose was new, so software did not exist for using it. Finally, the unknown quantities of most interest are the actual numbers of items with the feature in each of the lots (which were of known finite size), so the software must be able to sample posterior distributions of quantities which take on finitely many values.

Before introducing the statistical models, we begin with some notation. The finite population consists of M lots. Let the ith lot size be denoted by N_i and the unknown number of systems in the ith lot with the attribute be denoted by K_i. The sample sizes and numbers of sampled features in the convenience and random samples for the i the lot are denoted respectively by n_{ci}, y_{ci}, n_{ri} and y_{ri}. Because the convenience sample is assumed to be taken first, the ith lot size for the random sample is $N_{ri} = N_i - n_{ci}$ and $K_{ri} = K_i - y_{ci}$ is the number of components with the attribute remaining in the ith lot after the convenience sample has been taken. We assume

$$K_i \sim Binomial(N_i, p_i). \tag{1}$$

and

$$p_i \sim Beta(a, b). \tag{2}$$

Now we consider statistical models for the data. For the convenience sample data, we want to account for the potential bias of sampling too many or too few components with the attribute. To do this, we use the extended-hypergeometric distribution for y_{ci} which has probability function

$$P(y_{ci} = y) = \frac{\begin{pmatrix} n_{ci} \\ y \end{pmatrix} \begin{pmatrix} N_i - n_{ci} \\ K_i - y \end{pmatrix} \theta^y}{\sum_{j=\max(0,n_{ci}-N_i+K_i)}^{\min(n_{ci},K_i)} \begin{pmatrix} n_{ci} \\ j \end{pmatrix} \begin{pmatrix} N_i - n_{ci} \\ K_i - j \end{pmatrix} \theta^j} \ ,$$

for $y = \max(0, n_{ci} - N_i + K_i), \ldots, \min(n_{ci}, K_i)$. When the biasing parameter θ is equal to one, the extended-hypergeometric reduces to the hypergeometric which arises from a completely random sample in which there is no biasing. When θ is greater than one, the sampling favors components with the attribute. The randomly sampled data y_{ri} are assumed to follow hypergeometric distributions.

2.2 Example 2: System Reliability Based on Partially Informative Tests

Another reliability problem involves synthesizing two different types of data, neither of which is standard for reliability analysis. First, the system test data provide complicated information: for notational clarity, consider a single system test. If the set of components in the system is denoted by C, there is a subset of components C_1 that we know to have worked, another subset of components C_0 that we know to have failed, and a third subset of components C_2, at least one of which must have failed. (The test provides no information at all about the success or failure of the remaining components.) The system is a series system. The likelihood function for this single test, assuming that the system is of age t, is

$$\prod_{i \in C_1} p_i(t) \prod_{i \in C_0} \{1 - p_i(t)\} \left\{ 1 - \prod_{i \in C_2} p_i(t) \right\}, \tag{3}$$

where the first two products are defined to be one if empty, while the last is zero. Here $p_i(t)$ is the probability of success of component i at age t: we used $p_i(t) = \Phi\left\{(\sigma_i^2 + \gamma_i^2)^{-1/2}(\alpha_i + \beta_i t - \theta_i)\right\}$, where Φ is the Gaussian distribution function.

One reason for this choice of $p_i(t)$ (in particular, the seemingly redundant parameterization) is the other type of data we use: certain of the components were tested to assure that they met specifications, and these tests generate continuous data. If one assumes that the a specification measurement S_i on component i satisfies $S_i \sim N(\alpha_i + \beta_i t, \gamma_i^2)$, specification data can be incorporated naturally. Then if one assumes that, conditionally on its specification measurement S_i, the component would succeed in a system test with probability $\Phi\left\{\sigma_i^{-1}(S_i - \theta_i)\right\}$, it follows that unconditionally, the component's success probability is $\Phi\left\{(\sigma_i^2 + \gamma_i^2)^{-1/2}(\alpha_i + \beta_i t - \theta_i)\right\}$. More generally, suppose n_i specification measurements $S_{i1}, \ldots, S_{in_i}$ apply to component i, that $S_{ij} \sim N(\alpha_{ij} + \beta_{ij} t, \gamma_{ij}^2)$, and that the success probability given the specs is $\prod_{j=1}^{n_i} \Phi\{\sigma_{ij}^{-1}(S_{ij} - \theta_{ij})\}$. Then the unconditional success probability, or the likelihood function for the system tests, is $\prod_{j=1}^{n_i} \Phi\{(\sigma_{ij} + \gamma_{ij})^{-1/2}(\alpha_{ij} + \beta_{ij} t - \theta_{ij})\}$. We also generalize to multiple covariates; they need not be the same for different specs. A final complication, but one that is trivial to handle, is that the system is built in several different configurations: not all components are present in all configurations.

This formulation enables us to use specification data to help make inferences on parameters relevant to system tests. It also requires special-purpose software.

2.3 Example 3: Integrated System Reliability Based on Diverse Data

A fundamental problem of system reliability is estimating the reliability of a system whose components are combined in series and parallel subsystems, and where data relevant to the component qualities take on general (not necessarily binomial) forms. (The case of binomial data is discussed in Johnson *et al* (2003).) As a simple example, consider a three component series system. Binomial data are available on Component 1 at various ages, and the success probability at age t satisfies $\log(p_1(t)/\{1 - p_1(t)\}) = \alpha_0 - \alpha_1 t$. "Success" for Component 2 is defined in terms of its lifetime which is distributed Weibull: a lifetime η equates to a component success at time t if $t < \eta$, and data on Component 2 are a collection of possibly right-censored lifetimes. Component 3 is required to generate a desired amount of power τ on demand; the distribution of power is lognormal, with a logged mean that decreases linearly in age. Data are (power, age) pairs. Finally, we have binomial system test data, where the success probability is the age-dependent success probability of Component 1, multiplied by the reliability of Component 2, multiplied by the age-dependent probability of sufficient power generation by Component 3. The full data likelihood contains terms for each of the four types of tests, and other information can be captured in prior distributions. It is necessary that the software analyze likelihood functions with each of these terms, and ideally it would support the integration of components into (sub)systems in arbitrary parallel/series combinations.

As described above, the Component 1 data y_{1i} at time t_{1i} follow $Binomial(n_{1i}, p_{1i})$ where $\log(p_{1i}/(1 - p_{1i})) = \alpha_0 + \alpha_1 t_{1i}$. The Component 2 data y_{2i} follow $Weibull(\lambda, \beta)$ for scale λ and shape β. The Component 3 data y_{3i} at time t_{3i} follow $Lognormal(\mu_{3i}, \sigma^2)$, where $\mu_{3i} = \gamma_0 + \gamma t_{3i}$. Finally, the system data y_{si} at time t_{si} follow $Binomial(n_{si}, p_{si})$, where $p_{si} = R_1(t_{si})R_2(t_{si})R_3(t_{si})$ and $R_1(t_{si}) = \exp(\alpha_0 + \alpha_0 t_{si})/(1 + \exp(\alpha_0 + \alpha_0 t_{si}))$, $R_2(t_{si}) = exp(-\lambda t_{si}^{\beta})$ and $R_3(t_{si}) = 1 - \Phi\{(\log(\tau) - (\gamma_0 + \gamma t_{si})\}/\sigma)$.

3 YADAS: a Statistical Modeling Environment

YADAS is a software environment for performing arbitrary Markov chain Monte Carlo computations, and as such it is very useful for defining and analyzing new, nonstandard statistical models. Its source code and documentation, several examples, and supporting technical reports are available for download at `yadas.lanl.gov` (Graves, 2003a,b). Its software architecture makes it easy to define new terms in models and make small adjustments to existing models. MCMC algorithms often suffer from poor autocorrelation, and YADAS provides an environment for exploring and fixing these problems. YADAS is written in Java and generally requires additional Java code to work a new problem, but work continues on alternative interfaces. We discuss all these issues in this section.

3.1 Expressing Arbitrary Models

Defining a model in YADAS is as simple as specifying how to calculate the unnormalized posterior distribution evaluated at an arbitrary parameter value. This is an advantage of a Bayesian approach, as well as being one of the benefits of the design decision to emphasize the Metropolis–Hastings algorithm instead of Gibbs sampling as in WinBUGS (Spiegelhalter *et al*, 2000). In the Gibbs sampler, each time the model is changed, the sampling algorithm must be changed accordingly. In YADAS, however, the model definition is decoupled from the algorithm definition. Provided the acceptance probabilities are generated correctly, the distribution of the samples will converge to the desired posterior distribution. Since the acceptance probabilities are determined by the unnormalized posterior density function, this happens automatically when the new model is defined.

The definition of a model is a collection of objects called *bonds*. Each bond is a term in the posterior distribution. Bonds are defined in the software in such a way as to make it easy to make the sort of small changes to an analysis that are common in the model–building phase. Examples include changing a parameter from fixed to random, or changing a distributional form. In particular, analyzing the sensitivity of conclusions to the choice of prior is natural.

The ease of defining new models was particularly evident in the analysis of the reliability of the component manufactured in lots and sampled nonrandomly. The analysts needed to write code to calculate the extended hypergeometric density function, but after this trivial exercise was complete, it could be plugged in as any other density would be. Without YADAS, time constraints would have forced the analysts to make use of a more convenient but less appropriate analysis.

In our second example, the critical step was to compute (3) after first computing the $p_i(t)$'s. This was also straightforward, and the handling of specification data just required adding another bond (with the familiar normal linear model form) to the existing list. The third example, in which various forms of component test data are combined with system test data, is an excellent example of the usefulness of the YADAS model definition strategy. The component test data are as easy to include as in an application where they are the only data source. We make use of YADAS's general code that reads in a system structure and integrates component success probabilities in any series or parallel combination, in order to incorporate the system test data.

3.2 Special Algorithms

While it is true that defining an MCMC algorithm for a new problem is as easy as specifying how to compute the unnormalized posterior distribution, it is also true that these first attempts at algorithms may fail to perform adequately. However, YADAS turns this to a strength by helping users to improve algorithms by adding steps to the existing algorithm; Metropolis-Hastings based software is much better suited to this goal than Gibbs-based software. The most common MCMC performance problem is high posterior correlation among parameters; this generates high autocorrelation in consecutive MCMC samples, because parameters are reluctant to move individually. YADAS's typical approach is the "multiple parameter update": one proposes simultaneous moves to parameters in a direction of high variability. For example, in our second example (as happens in many generalized linear model examples), the intercept and slope parameters for some components were highly correlated, and the algorithm was improved with new steps that proposed the addition of a random amount to the intercept while simultaneously subtracting a multiple of the same amount from the slope.

The naturalness of defining algorithms in YADAS was also exhibited in the biased sampling problem, where the numbers of items in each lot with the feature of interest needed to be sampled; this was handled with YADAS's general approach for sampling parameters that take on finitely many values.

3.3 Interfaces, Present and Future

YADAS is written in Java, and that provides portability advantages beyond its encouragement of generality that helped YADAS become as ambitious as it is. However, few Bayesian statisticians use Java as their language of choice, so this limits its popularity. An area of active YADAS development is providing additional interfaces to its capabilities. One such interface is the interface with the R package (`www.R-project.org`), a very popular, free statistics computing environment that is very similar to S-Plus. This interface will facilitate the handling of both input to and output from MCMC algorithms, including examining output for adequate convergence to the limiting distribution and rapid mixing.

One application is the use of genetic algorithms for experiment design: each candidate design selected by the genetic algorithm will generate data, which will then be analyzed using YADAS, and the analysis results will be examined for "fitness" and fed back to determine the next genetic algorithm generation. System reliability is an application of particular interest. For a given budget, which system, subsystem or component data be collected to reduce the uncertainty of system reliability the most? (See Hamada *et al* (2004) for an example involving a fault tree.) The R-YADAS interface is possible thanks to the SJava package of the omegahat project (`www.omegahat.org`).

Another interface that will particularly help with reliability problems is the interface with a new graphical tool for eliciting defining system structure and its relationship to data (Klamann and Koehler, 2004).

4 Examples revisited

4.1 Example 1

We illustrate the problem with convenience and random samples from the stratified population with a simulated data set; the data from the real application are proprietary. The simulated data set features a total population of 5000 items in 230 lots (100 lots of size 10, 100 of size 25, and 30 of size 50). Total sample sizes were 100 for the convenience sample and 50 for the random sample, with 18 lots being sampled in both ways, 21 only randomly sampled, and 57 only sampled by convenience. A total of 513 components had the feature: individual lot feature proportions p_i were drawn from the Beta$(1, 9)$ distribution, and the true value of θ was 2, representing mild biasing. In the convenience sample, 16 (16%) had the feature, while 6 features appeared in the random sample (12%). A naive estimate for the feature prevalence would then be $22/150 = 0.147$. 10.1% of the unsampled items had the feature, so the feature was overrepresented in both samples.

We used the following prior distributions: $\frac{a}{a+b} \sim Beta(0.3, 1.7)$, $a + b \sim Gamma(2, 5)$, and $\log\theta \sim N(0, 1)$. Posterior medians for these quantities were then 0.13, 5.5, and (for θ) 1.32. The posterior median for p^*, the fraction of unsampled items with the feature, was 0.13, different from the naive estimate and closer to the random sample fraction than the convenience sample fraction, as it should be. A 90% posterior interval for this quantity is $(0.073, 0.209)$.

4.2 Example 2

The system we studied in Example 2 had a total of 23 components. Thirteen of these had no related specification measurements, five had a single specification, and five had between two and four. Roughly 1000 system tests were available, with a proprietary number of failures. The twenty specification measurements generated a total of roughly 2000 data points. We studied the effects of six covariates, and the posterior distribution involved 294 unknown parameters.

Using YADAS to analyze these data required us to write code to handle the system test

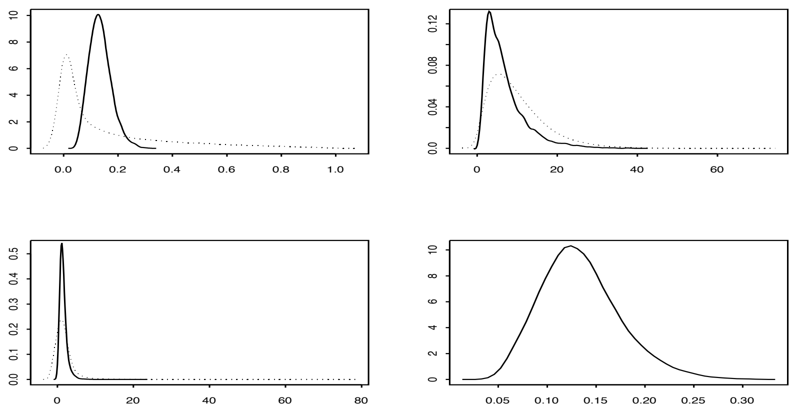

Figure 1: Plot of Example 1 Parameter Posteriors. Dotted curves represent priors. The four plots are for $a/(a+b)$, $a+b$, θ, and p^*.

data in the form of successes, failures, and partial successes. We needed to calculate the likelihood for each system test, and all the covariates affected the success probabilities for each component in a novel way. Also, many combinations of the 294 unknown parameters were highly correlated due to shortcomings in the completeness of the data, and this led to poorly performing algorithms. However, YADAS made it possible to improve these algorithms using multiple parameter updates. Results are shown in Figure 2; these are reliability posterior distributions for four different versions of the system. The axes are not shown because of the sensitivity of the data.

4.3 Example 3

The system pass/fail data consisted of 15 tests at each of 0, 5, 10, 15 and 20 years. The Component 1 pass/fail data consisted of 25 tests at each of 0, 2, 4, 6, 8, 10, 15 and 20 years. The Component 2 consisted of 25 lifetimes. Finally, the Component 3 data consisted of 10 destructive observations at each of 0, 2.5, 5, 7.5, 10, 15 and 20 years. Note that some of the system and component data are collected at different rates.

Analyzing these data provide the system reliability median and 95% credible interval over time as displayed in Figure 3. The component model parameter posteriors are given in

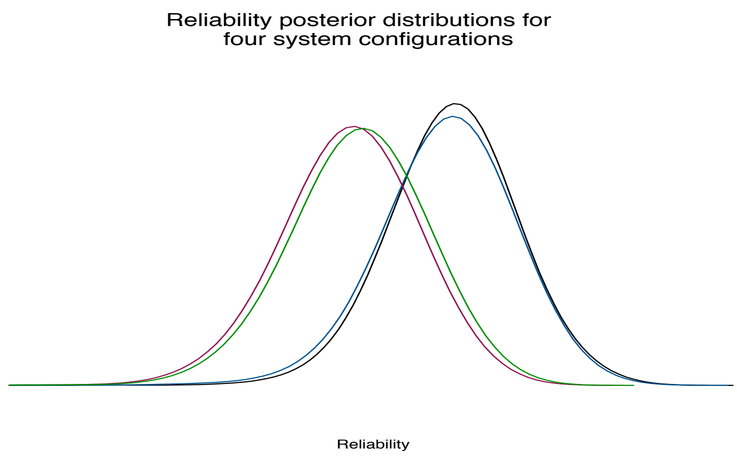

Figure 2: Density estimates for four different versions of the system described in Example 2 and for one value of the covariates. The axes are omitted due to proprietary concerns.

Figures 4 and 5. Note that the posteriors plotted as dotted lines did not use the system data. Those posteriors plotted as solid lines which use the system data are tighter and therefore more informative. The system data provided little additional information for the Component 3 model parameters as displayed in Figure 5. Figure 6 displays the system reliability 95% credible interval over time when both the system data are used and when they are not used. The solid lines display the results when the system data are used. Note that the solid lines are higher and closer together than the dotted lines.

5 Discussion

In this paper we have attempted to communicate some of the excitement of working on modern system reliability assessments. New methodology is important for dealing with such issues as nonrandom sampling, and analyzing test results that utilize different levels of the system and generate different data distributions. Though we have not illustrated them in this paper, other forms of information such as expert judgment, more detailed engineering models, and simulation models also need to be integrated with traditional data, and there are opportunities for research into good methods for doing this. Appropriate new models require new computational methods, and an extensible modeling environment makes it practical to

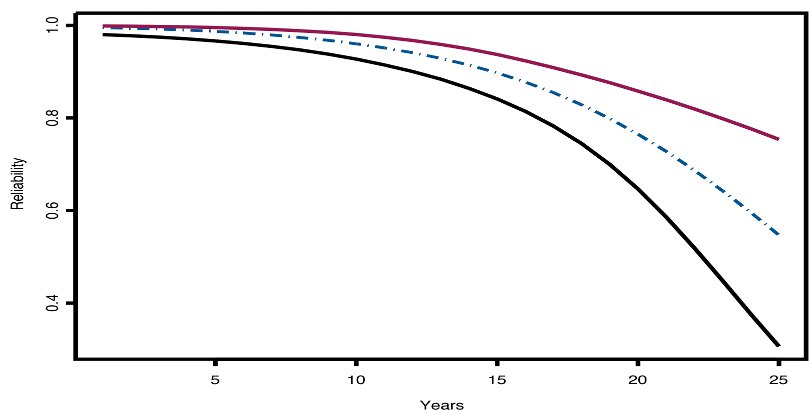

Figure 3: Plot of Example 3 System Reliability Posterior Median and 95% Credible Interval.

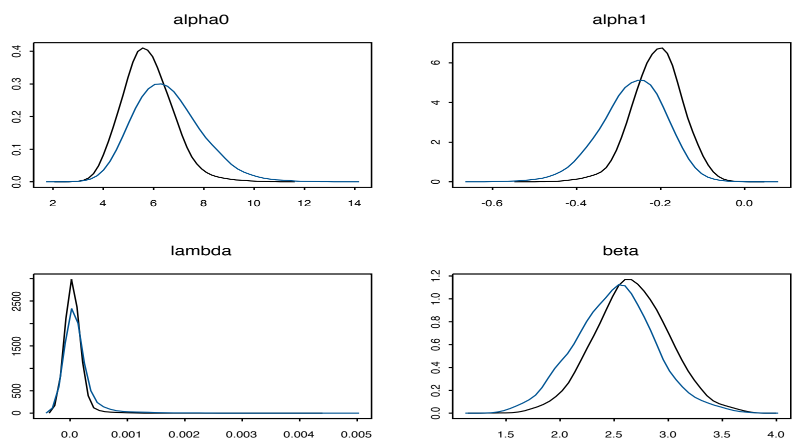

Figure 4: Plot of Example 3 Component Models 1 and 2 Parameter Posteriors (with system data (solid line) and without system data (dotted line)).

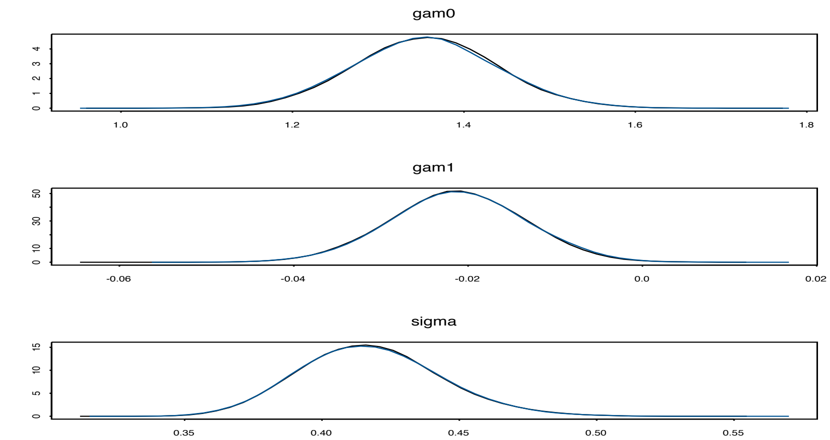

Figure 5: Plot of Example 3 Component Model 3 Parameter Posteriors (with system data (solid line) and without system data (dotted line)).

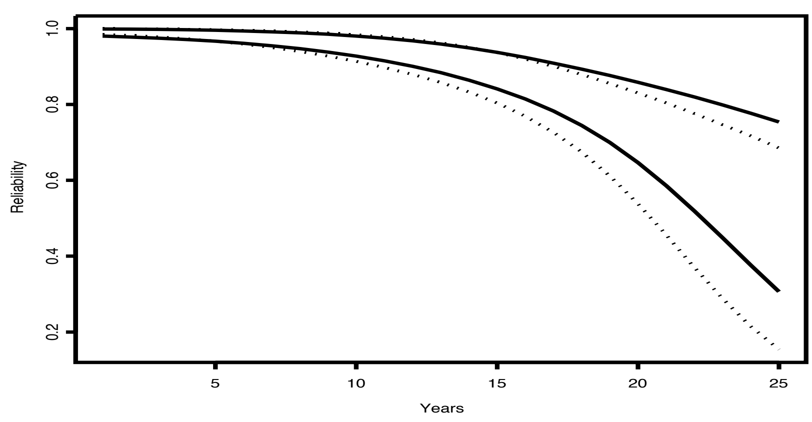

Figure 6: Plot of Example 3 System Reliability Posterior Posterior 95% Credible Intervals (with system data (solid line) and without system data (dotted line)).

work with new models, even in a deadline–driven scientific environment. Finally, complex system models with automatic analysis software are amenable to exciting research on using genetic algorithms to guide resource allocation and experimental design.

Acknowledgments

We thank Dee Won for her encouragement of this work.

References

Graves, T.L. (2003a). A framework for expressing and estimating arbitrary statistical models using Markov chain Monte Carlo. Los Alamos National Laboratory Technical Report LA-UR-03-5934.

Graves, T.L. (2003b). An introduction to YADAS. `yadas.lanl.gov`.

Graves, T., Hamada, M., Booker, J., Decroix, M., Bowyer, C., Chilcoat, K., Thompson, S.K. (2004). Estimating a proportion using stratified data arising from both convenience and random samples. Los Alamos National Lab Technical Report LA-UR-03-8396.

Hamada, M., Martz, H.F., Reese, C.S., Graves, T., Johnson, V., Wilson, A.G. (2004). A fully Bayesian approach for combining multilevel failure information in fault tree quantification and optimal follow-on resource allocation. *Reliability Engineering and System Safety*, 86, 297-305.

Johnson, V., Graves, T., Hamada, M., Reese, C.S. (2003). A hierarchical model for estimating the reliability of complex systems (with discussion). *Bayesian Statistics 7*, Oxford University Press, 199-213, Bernardo, J.M., Bayarri, M.J., Berger, J., Dawid, A.P., Heckerman, D., Smith, A.F.M. and West, M. (Eds.).

Cole, P.V.Z. (1975). A Bayesian reliability assessment of complex systems for binomial sampling. *IEEE Transactions on Reliability*, R-24, 114-117.

Klamann, R. and Koehler, A. (2004). GROMIT: Graphical Modeling Tool for System Statistical Structure. Los Alamos, NM: Los Alamos National Laboratory.

Martz, H.F., Waller, R.A. (1990). Bayesian reliability anaysis of complex series/parallel systems of binomial subsystems and components. *Technometrics*, 32, 407-416.

Martz, H.F., Waller, R.A., Fickas, E.T. (1988). Bayesian reliability analysis of series systems of binomial subsystems and componenents. *Technometrics*, 30, 143-154.

Mastran, D.V. (1976). Incorporating component and system test data into the same assessment: a Bayesian approach. *Operations Research*, 24, 491-499.

Mastran, D.V., Singpurwalla, N.D. (1978). A Bayesian estimation of the reliability of coherent structures. *Operations Research*, 26, 663-672.

Natvig, B., Eide, H. (1987). Bayesian estimation of system reliability. *Scandinavian Journal of Statistics*, 14, 319-327.

Spiegelhalter, D., Thomas, A., Best, N. (2000). *WinBUGS Version 1.3 User Manual.*